A Datasheet

The following is the datasheet (Gebru et al., 2021) for Off-the-Grid Multi-Agent Reinforcement Learning (OG-MARL).

Note: The OG-MARL repository will be released on GitHub after the anonymous review. For now the OG-MARL datasets and code is downloadable from our anonymised website:

https://sites.google.com/view/og-marl

A.1 Motivation

For what purpose was the dataset created? The datasets in OG-MARL were created to facilitate research in offline Multi-Agent Reinforcement Learning (MARL). Offline MARL is a nascent field of machine learning that promises to unlock real-world applications of MARL. However, progress has been hampered by the lack of a standardised, high-quality benchmark datasets. OG-MARL was built to fill this gap and drive progress in the field.

Who created the dataset and on behalf of which entity? OG-MARL was created by <anonymous> on behalf of <anonymous> and <anonymous>.

Who funded the creation of the dataset? The creation of OG-MARL was funded by <anonymous>.

A.2 Composition

What do the instances that comprise the dataset represent? The various datasets in OG-MARL comprise of environment transitions in popular MARL benchmark environments (e.g. SMAC by Samvelyan et al. (2019)). The transitions were generated by recording environment interactions between policies trained using online RL.

How many instances are there in total? Each dataset in OG-MARL has approximately 1 million transitions in it.

Does the dataset contain all possible instances or is it a sample of instances from a larger set? Great care was taken to ensure that the dataset in OG-MARL had good coverage of the state and action space of the environment. It is however, not possible (nor desirable) to guarantee full coverage.

What data does each instance consist of? Each instance consists of a sequence of multi-agent transitions in the environment. A transition is composed of agent observations, actions, rewards and next observations $(\{o_t^1, \dots, o_t^n\}, \{a_t^1, \dots, a_t^n\}, \{r_t^1, \dots, r_t^n\}, \{o_{t+1}^1, \dots, o_{t+1}^n\})$.

Is there a label or target associated with each instance? As we are in the reinforcement learning paradigm, instances do not have *labels*. However, since each instance is a multi-agent transition, they do each have a corresponding reward for each agent, which we use for training.

Is any information missing from individual instances? Everything is included. No data is missing.

Are relationships between individual instances made explicit? Transitions that belong to the same episode can be retrieved together, if desired.

Are there recommended data splits? In offline RL one does not need to split data like in supervised learning. All data can be used for training.

Are there any errors, sources of noise, or redundancies in the dataset? None.

Is the dataset self-contained, or does it link to or otherwise rely on external resources? OG-MARL is completely self-contained. The datasets are stored in a binary format but can be loaded into a dataset loader with the utilities provided in the OG-MARL code.

Does the dataset contain data that might be considered confidential? No.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? No.

Does the dataset identify any subpopulations? No.

Is it possible to identify individuals, either directly or indirectly from the dataset? No.

Does the dataset contain data that might be considered sensitive in any way? No.

A.3 Collection Process

How was the data associated with each instance acquired? To generate the datasets for OG-MARL we trained online MARL algorithms on a variety of popular MARL benchmark environments and recorded the environment transitions.

What mechanisms or procedures were used to collect the data? We trained our online MARL algorithms on a PC with a GPU (Nvidia RTX 3070) and recorded experiences with a python utility we designed and subsequently open-sourced to the community.

If the dataset is a sample from a larger set, what was the sampling strategy? The different datasets in OG-MARL have different data compositions. We grouped transitions into `Good`, `Medium` and `Poor` according to the return of episode that the transition belonged to.

Who was involved in the data collection process? <Anonymous> and <anonymous>.

Over what timeframe was the data collected? The datasets in the current version of OG-MARL were collected over a period of about 3 months.

Were any ethical review processes conducted? No, since it is believed that none was required.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources No.

Were the individuals in question notified about the data collection? Not applicable.

Did the individuals in question consent to the collection and use of their data? Not applicable.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? Not applicable.

Has an analysis of the potential impact of the dataset and its use on data subjects been conducted? Not applicable.

A.4 Preprocessing/Cleaning/Labeling

Was any preprocessing/cleaning/labeling of the data done? Transitions were grouped into short continuous sequences to allow for training recurrent policies.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data? Individual transitions can be loaded instead of sequences.

Is the software that was used to preprocess/clean/label the data available? Yes, in the OG-MARL repository.

A.5 Uses

Has the dataset been used for any tasks already? Yes. Zhu et al. (2023) and Formanek et al. (2023) both used OG-MARL.

Is there a repository that links to any or all papers or systems that use the dataset? There is a repository hosted by <anonymous> at <anonymous>.

What (other) tasks could the dataset be used for? OG-MARL could be used for any kind of sequential decision-making research.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? Loading entire episodes of transitions needs to be made easier in future releases.

Are there tasks for which the dataset should not be used? The data in OG-MARL was generated on simplified environments and does not necessarily generalise to the real world.

A.6 Distribution

Will the dataset be distributed to third parties outside of the entity on behalf of which the dataset was created? Yes, OG-MARL is publicly available on the internet.

How will the dataset will be distributed? OG-MARL datasets are hosted in an S3 bucket but can easily be accessed via our publicly open website or by running the download scripts in the OG-MARL code.

When will the dataset be distributed? The datasets were released in February of 2023.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? We have applied the *CC BY-NC-SA* dataset licence to OG-MARL.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? No.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? No.

A.7 Maintenance

Who will be supporting/hosting/maintaining the dataset? <Anonymous> will be responsible for maintaining the datasets on behalf of <anonymous>, who will also be financially supporting the hosting of the datasets.

How can the owner/curator/manager of the dataset be contacted? Via email, <anonymous>.

Is there an erratum? Versioning and changes are tracked on the OG-MARL repository, <anonymous>.

Will the dataset be updated? OG-MARL is a growing collection of offline MARL datasets. The creator and the wider community will be adding new datasets over time.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances? Not applicable.

Will older versions of the dataset continue to be supported/hosted/maintained? Yes.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? Yes, please open a pull request on the OG-MARL repository.

B Additional Environment Information

In this section we provide additional information of all of the environments supported in OG-MARL. In Table B.1 we provide an overview of salient environment characteristics, in addition to the algorithm which was used to generate the behaviour policies. In Table B.2 we provide links to the source of the environments for the reader to refer to for additional information about the environments.

Table B.1: All supported environments and scenarios in OG-MARL and some of their characteristics.

Environment	Scenario	Agents	Actions	Observations	Reward	Agent Types	Behaviour	Online Perf.
SMAC	3m	3	Discrete	Vector	Dense	Homog	QMIX	16.1
	8m	8				Homog		16.2
	2s3z	5				Heterog		18.2
	5m_vs_6m	5				Homog		16.6
	27m_vs_30m	27				Homog		16.0
	3s5z_vs_3s6z	8				Heterog		17.0
	2c_vs_64zg	2				Homog		18.0
MAMuJoCo	2-HalfCheetah	2	Continuous	Vector	Dense	Heterog	MATD3	6924
	2-Ant	2				Homog		2621
	4-Ant	4				Homog		2769
PettingZoo	Pursuit	8	Discrete	Pixels	Dense	Homog	QMIX	79.5
	Co-op Pong	2	Discrete			Heterog	IDQN	65.1
	Pistonball	15	Continuous			Homog	MATD3	84.6
Flatland	3 Trains	3	Discrete	Vector	Dense	Homog	IDQN	-5.1
	5 Trains	5						-5.9
SMAC v2	terran_5_vs_5	5	Discrete	Vector	Dense	Hetrog	QMIX	17.0
	zerg_5_vs_5	5						15.2
	terran_10_vs_10	10						16.9
CityLearn	2022_all_phases	17	Continuous	Vector	Dense	Homog	ITD3	-6421
Voltage Control	case33_3min_final	6	Continuous	Vector	Dense	Homog	ITD3	-12.3

Table B.2: All environments with links to their sources.

Environment	**Website**
SMAC v1	https://github.com/oxwhirl/smac
SMAC v2	https://github.com/oxwhirl/smacv2
PettingZoo	https://pettingzoo.farama.org/
Flatland	https://flatland.aicrowd.com/intro.html
MAMuJoCo	https://github.com/schroederdewitt/multiagent_mujoco
CityLearn	https://github.com/intelligent-environments-lab/CityLearn
Voltage Control	https://github.com/Future-Power-Networks/MAPDN

C Additional Information on Datasets

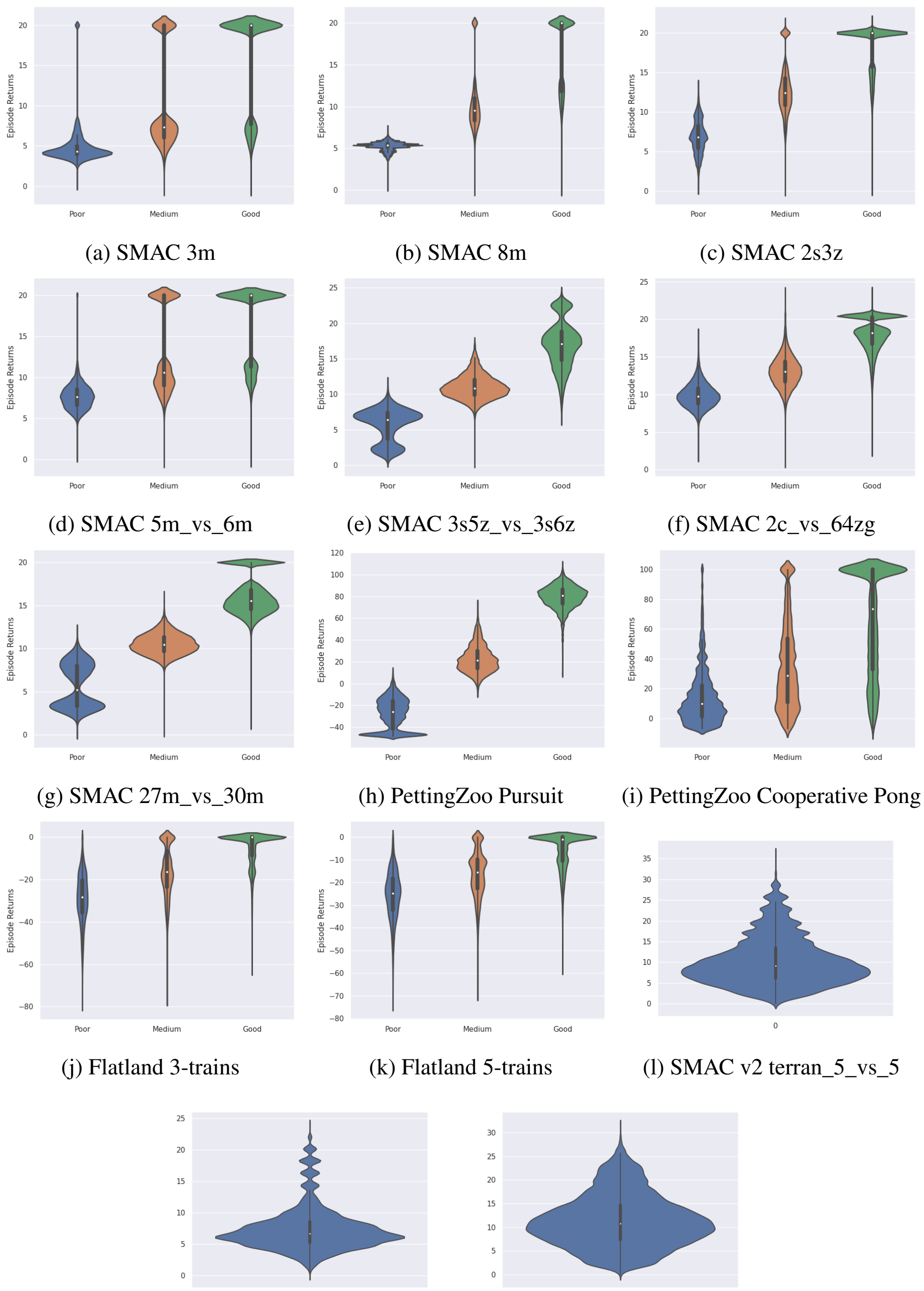
In this section, we provide additional information about the datasets in OG-MARL. In Table C.1 we give the mean episode return with standard deviation for all datasets in OG-MARL.

Table C.1: Table of the mean episode return with the standard deviation for all datasets in OG-MARL.

Environment	Scenario	Dataset	Mean Episode Return (± Std)	Number of Sequences
SMAC	3m	Good	16.0±6.1	120569
		Medium	10.0±6.0	120004
		Poor	4.8±2.3	118447
	8m	Good	16.3±4.4	111873
		Medium	10.3±3.4	120845
		Poor	5.3±0.6	109515
	5m_vs_6m	Good	16.6±4.7	112779
		Medium	12.8±5.1	117594
		Poor	7.7±1.5	110031
	2s3z	Good	18.2±2.9	107900
		Medium	12.8±3.1	107640
		Poor	6.8±2.1	101197
	3s5z_vs3s6z	Good	17.0±3.3	101335
		Medium	11.0±1.7	107873
		Poor	5.7±2.3	107475
	2c_vs_64zg	Good	18.0±2.2	108270
		Medium	13.1±2.0	111199
		Poor	9.9±1.6	115370
	27m_vs_30m	Good	16.0±2.1	110271
		Medium	10.5±1.2	113737
		Poor	5.7±2.5	110845
MAMuJoCo	2-HalfCheetah	Good	6924±1270	100000
		Medium	1484±469	100000
		Poor	400±333	100000
	2-Ant	Good	2621±493	100041
		Medium	1099±264	100109
		Poor	437±164	99804
	4-Ant	Good	2769±270	100170
		Medium	1546±389	100215
		Poor	542±216	100224
PettingZoo	Pursuit	Good	79.5±10.8	101249
		Medium	22.7±12.4	100087
		Poor	-27.3±14.0	100000
	Co-op Pong	Good	65.1±35.6	100687
		Medium	35.6±29.9	101490
		Poor	14.4±18.7	102277
	Pistonball	Good	84.6±17.9	208518
		Medium	34.1±25.6	200142
		Poor	12.0±22.6	200000
Flatland	3 Trains	Good	-5.2±8.0	23000
		Medium	-16.1±11.8	19800
		Poor	-28.8±11.8	19200
	5 Trains	Good	-5.9±8.0	20600
		Medium	-16.3±10.2	18000
		Poor	-25.5±10.5	17600
SMAC v2	terran_5_vs_5	Replay	10.4±5.9	97795
	zerg_5_vs_5	Replay	7.5±3.6	137776
	terran_10_vs_10	Replay	11.4±5.6	75355
CityLearn	2022_all_phases	Replay	-6820.7±458.4	169068
Voltage Control	case33_3min_final	Replay	-25.1±22.3	40541

678 C.1 Violin Plots

In addition to the table with mean episode returns, we also provide violin plots for all datasets in OG-MARL in order to visualise the distribution of episode returns induced by the behaviour policies.



(a) SMAC 3m (b) SMAC 8m (c) SMAC 2s3z

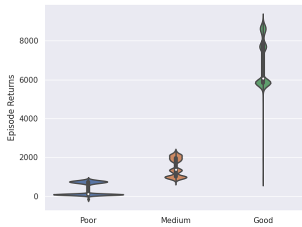
(d) SMAC 5m_vs_6m (e) SMAC 3s5z_vs_3s6z (f) SMAC 2c_vs_64zg

(g) SMAC 27m_vs_30m (h) PettingZoo Pursuit (i) PettingZoo Cooperative Pong

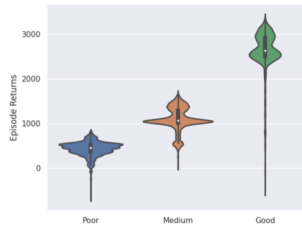
(j) Flatland 3-trains (k) Flatland 5-trains (l) SMAC v2 terran_5_vs_5

(m) SMAC v2 zerg_5_vs_5 (n) SMAC v2 terran_10_vs_10

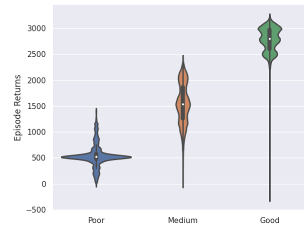
Figure C.1: Violin plots of all datasets with discrete actions.



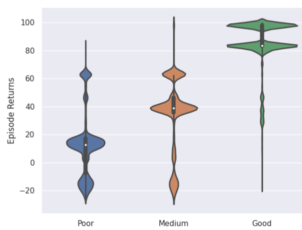
(a) MAMuJoCo 2-halfcheetah



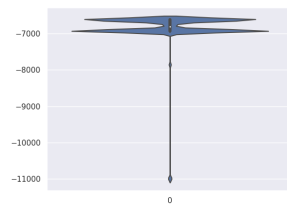
(b) MAMuJoCo 2-ant



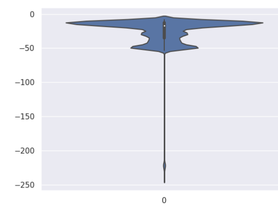
(c) MAMuJoCo 4-ant



(d) PettingZoo Pistonball



(e) CityLearn



(f) Voltage Control

Figure C.2: Violin plots of all datasets with continuous actions.

D Additional Baseline Information

In this section, we provide additional implementation details for each of the baseline algorithms implemented in OG-MARL as well as the hyper-parameters used for the experiments and additional results.

D.1 Background: Single Agent Offline RL Algorithms

As mentioned in the main text, the primary challenge algorithms need to address during offline training is data distribution mismatch between the behaviour (offline) data and the induced online data. For example, the state visitation frequency induced by the behaviour policy is typically different to that of the learnt policy. While state distribution mismatch can cause failure when the algorithm is deployed, it does not generally cause any issues during training, and can easily be mitigated by expanding the breadth and diversity of the dataset (Agarwal et al., 2019). On the other hand, the most common and difficult-to-address type of distribution mismatch in offline RL is out-of-distribution (OOD) actions. An offline RL algorithm may assign a high value to an OOD action during training due to the extrapolation done by the neural network (Fujimoto et al., 2019). These errors then tend to propagate to other state-action pairs, as Q-learning and related algorithms use bootstrapping to compute Bellman targets (Kumar et al., 2019). The propagation of extrapolation error then manifests itself as a kind of "unlearning", where the performance of the offline RL algorithm rapidly starts to degrade with further training. Most of the remedies proposed in the literature to address OOD actions can be grouped into one of two categories.

Policy constraints. Several methods try to resitrict the degree to which the learnt policy can become off-policy with respect to the behavioural policy. These methods tend to incorporate some form of behaviour cloning (BC) into RL algorithms to force the learnt policy to remain relatively online with respect to the behaviour dataset. *Batch-Constrained Q-learning* (BCQ) (Fujimoto et al., 2019) and *Twin Delayed DDPG + behaviour cloning* (TD3 + BC) (Fujimoto and Gu, 2021) are two popular algorithms in this class.

Conservative value regularisation. The second approach mitigates extrapolation error by regularising the learnt value function to avoid overestimating values for OOD actions. An example of this approach, called *conservative Q-learning* (CQL), has been successfully applied to Q-learning and actor-critic methods by Kumar et al. (2020) in single-agent offline RL.

D.2 Multi-Agent Offline MARL Algorithms

At the time of writing, there are only a handful of cooperative offline MARL algorithms available in the literature and we endeavoured to implement as many of them as possible in OG-MARL. However, several algorithms proposed in the literature do not have open-source implementations online and were therefore challenging to re-implement. In Table D.1 we give an overview of all the algorithms in the literature and whether we re-implemented them in OG-MARL.

D.3 Implementation Details

In this section, we highlight the most important implementation details for the algorithms in OG-MARL and refer the reader to our open-source code for finer details.

QMIX. Our QMIX implementation is very similar to the original (Rashid et al., 2018). We use a single shared Q-network for all agents and concatenate agent IDs to the agent observations so that the network can distinguish between different agents. As in the original QMIX paper, our Q-network is a recurrent network that takes independent agent observations as input, while the mixing network conditions on global state information. To improve the performance of our QMIX implementation, we adopt the recommendation from Hu et al. (2021) to use *Q-lambda* (Sutton and Barto, 2018) to compute target Q-values.

Table D.1: An overview of cooperative offline MARL algorithms from the literature grouped by the work that proposed them as a novel algorithm or baseline. In the second column, we indicate if the code for the algorithm was originally made available online (open-sourced) and in the third column we indicate if the algorithm is implemented in OG-MARL. Algorithms in bold were the main contribution of the respective work while the rest are baselines used in the work. QMIX+CQL and QMIX+BCQ are novel baselines proposed in this work.

Algorithm Name	Open-Sourced	OG-MARL
MABCQ	✗	✗
MAICQ	✓	✓
DOP+CQL	✗	✗
DOP+BCQ	✗	✗
OMAR	✓	✓
ITD3+CQL	✓	✓
ITD3+BC	✗	✓
MATD3+CQL	✗	✓
MATD3+BC	✗	✓
QMIX+CQL	n/a	✓
QMIX+BCQ	n/a	✓

QMIX+CQL. We add conservative Q-learning (Kumar et al., 2020) to QMIX by uniformly sampling a number of joint-actions from the joint-action space and using those to select Q-values before passing them through the mixing network and using the resulting *mixed* Q-values to calculate the CQL-loss term.

QMIX+BCQ. We add discrete BCQ (Fujimoto et al., 2019) to QMIX by additionally training a behaviour cloning policy which we use to evaluate how likely each action is to be taken by the behaviour policy given the dataset. If the likelihood is below some threshold, we mask out that action during Q-learning in QMIX.

MAICQ. Our MAICQ implementation is as close as possible to the original by Yang et al. (2021).

ITD3 and MATD3. Our ITD3 and MATD3 use a shared policy network and shared Q-network, and concatenate agent IDs to agent observations. The policy is a recurrent neural network with a single GRU layer while the critic is a feedforward neural network that takes the global state as input instead of the observations.

ITD3+BC and MATD3+BC. We incorporate behaviour cloning into ITD3 and MATD3 by adding a behaviour cloning term to the policy learning step as in Fujimoto and Gu (2021)

ITD3+CQL and MATD3+CQL. We incorporate conservative Q-learning into ITD3 and MATD3 in a very similar way to how it was done by Pan et al. (2022).

OMAR We tried to keep our implementation of OMAR as close to the original (Pan et al., 2022) as possible. The main difference in our implementation is that the policy is a recurrent network, while in the original, they used a feedforward network.

D.4 Hyper-Parameters

In this section, we highlight the values we used for the most important hyper-parameters in our benchmark experiments. For additional details about the hyper-parameters we used, we refer to the `experiments` directory in our open-source code. In Table D.2 and Table D.3 we give the hyper-parameters for SMAC and MAMuJoCo experiments respectively. In order to keep the online evaluation budget fixed (Kurenkov and Kolesnikov, 2022) we tuned hyperparameters on *3m* and *2-Agent HalfCheetah* for SMAC and MAMuJoCo respectively.

Table D.2: Hyper-Parameters for Discrete Action Algorithms.

Algorithm	**Hyper-Parameter Name**	**Value**
All	Batch Size	32
	Optimiser	Adam
	Learning Rate	1e-3
	Hidden Activation Function	ReLu
	Q-Lambda	0.6
BC	Policy Linear Layer Dimension	64
	Policy GRU Layer Dimension	64
QMIX	Q-Network Linear Layer Dimension	64
	Q-Network Linear Layers Dimension	64
	Hyper-Network Dimension	64
	Mixing Embedding Dimension	32
	Soft Target Update Rate	1e-2
QMIX+BCQ	QMIX Hyper-Parameters	Same as above.
	Behaviour Network Linear Layer Dimension	64
	Behaviour Network GRU Layer Dimension	64
	Behaviour Threshold	0.4
QMIX+CQL	QMIX Hyper-Parameters	Same as above.
	CQL Alpha	2.0
	Number of Sampled Actions	20
MAICQ	Policy Network Linear Layer Dimension	64
	Policy Network GRU Layer Dimension	64
	Critic Network First Linear Layer Dimension	64
	Critic Network Second Linear Layer Dimension	64
	Mixing Hyper-Network Dimension	64
	Mixing Embedding Dimension	64
	MAICQ Epsilon	0.5
	MAICQ Advantages Beta	0.1
	MAICQ Target Q-Taken Beta	1000

Table D.3: Hyper-Parameters for Continuous Action Algorithms.

Algorithm	**Hyper-Parameter Name**	**Value**
All	Batch Size	32
	Optimiser	Adam
	Learning Rate	5e-4
	Hidden Activation Function	ReLu
	Policy Linear Layer Dimension	128
	Policy GRU Layer Dimension	128
ITD3	Critic Linear Layer Dimension	128
	Critic Linear Layers Dimension	128
	Target Update Rate	0.01
ITD3+BC	Behaviour Cloning Alpha	2.5
ITD3+CQL	CQL Alpha	10.0
	Number of OOD Actions	10
OMAR	CQL Parameters	Same as above.
	OMAR Iterations	3
	OMAR Number of Samples	20
	OMAR Number of Elites	5
	OMAR Sigma	2.0
	OMAR Coefficient	0.7

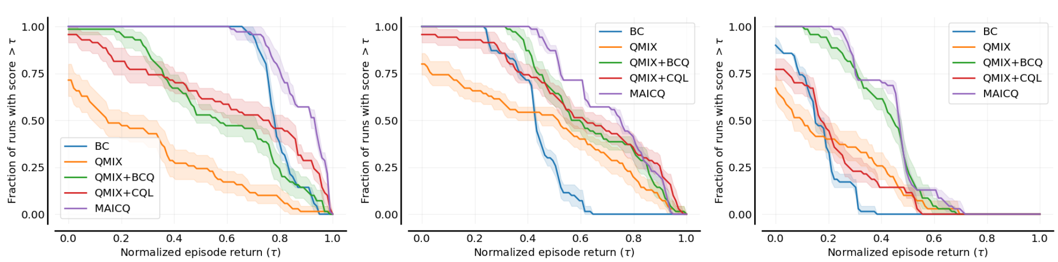
D.5 Additional Results

In this section, we provide additional baseline results on datasets in OG-MARL.

Discrete Actions. In Table D.4 we give the baseline results on datasets with discrete actions. In Figure D.1 we provide the aggregated performance profiles for SMAC.

Table D.4: Baseline results on datasets with discrete actions. The mean episode return with one standard deviation across all seeds is given. The best mean episode return in each row is given in bold.

Environment	Scenario	Dataset	BC	QMIX	QMIX+BCQ	QMIX+CQL	MAICQ
SMAC	3m	Good	16.0±1.0	13.8±4.5	16.3±1.5	**19.6±0.3**	18.8±0.6
		Medium	8.2±0.8	17.3±0.9	18.3±1.2	**18.9±1.2**	18.1±0.7
		Poor	4.4±0.1	10.0±2.9	12.4±2.3	5.8±0.4	**14.4±1.2**
	8m	Good	16.7±0.4	4.6±2.8	12.7±6.3	11.3±6.1	**19.6±0.3**
		Medium	10.7±05	13.9±1.6	16.0±1.4	16.8±3.1	**18.6±0.5**
		Poor	5.3±0.1	6.0±1.3	5.8±1.4	4.6±2.4	**10.8±0.8**
	5m_vs_6m	Good	**16.6±0.6**	8.0±0.5	8.3±0.9	13.8±3.9	16.3±0.9
		Medium	12.4±0.9	11.9±1.1	12.1±1.3	**16.9±1.2**	15.3±0.7
		Poor	7.5±0.2	10.7±0.9	**11.0±0.9**	10.4±1.0	9.4±0.4
	27m_vs_30m	Good	15.7±0.3	3.2±1.4	10.2±1.4	6.0±3.3	**16.1±1.8**
		Medium	10.3±0.4	6.2±2.1	9.8±1.2	8.0±1.7	**12.9±0.5**
		Poor	6.0±1.5	2.1±1.7	**10.3±0.7**	3.7±2.7	10.1±0.8
	2s3z	Good	18.2±0.4	5.9±3.4	16.6±1.2	19.0±0.8	**19.6±0.3**
		Medium	12.3±0.7	5.2±0.9	13.6±1.5	14.3±2.0	**17.2±0.6**
		Poor	6.7±0.3	3.8±1.2	11.5±1.0	10.1±0.7	**12.1±0.4**
	3s5z_vs_3s6z	Good	15.0±0.6	3.1±1.3	8.4±0.7	7.3±1.9	**16.2±0.7**
		Medium	10.6±0.2	3.0±1.0	10.5±0.8	8.1±3.1	**12.3±0.3**
		Poor	6.1±0.3	2.8±1.0	8.2±0.9	2.9±0.9	**8.4±0.2**
	2c_vs_64zg	Good	17.5±0.4	10.9±4.0	18.7±0.8	18.1±0.8	**19.3±0.3**
		Medium	12.5±0.3	16.8±1.6	**18.4±0.5**	14.9±0.7	14.6±0.6
		Poor	9.7±0.2	11.6±2.2	**14.3±0.8**	12.1±0.4	12.5±0.4
PettingZoo	Co-op Pong	Good	31.2±3.5	0.6±3.5	1.9±1.1	**90.0±4.7**	75.4±3.9
		Medium	21.6±4.8	10.6±17.6	20.3±12.2	64.9±15.0	**84.6±0.9**
		Poor	1.0±0.9	14.4±16.0	30.2±20.7	52.7±8.5	**74.8±7.8**
	Pursuit	Good	78.3±1.8	6.7±19.0	66.9±14.0	54.4±6.3	**92.7±3.7**
		Medium	15.0±1.6	-24.4±20.2	16.6±10.7	20.6±10.3	**35.3±3.0**
		Poor	-18.5±1.6	-43.7±5.6	**-0.7±4.0**	-19.6±3.3	-4.1±0.7
Flatland	3 Trains	Good	-5.6±2.4	-3.6±0.4	-3.5±2.8	**-2.1±0.4**	-25.3±0.2
		Medium	**-4.5±2.5**	-12.5±1.0	-7.1±2.7	-4.6±0.5	-25.2±0.2
		Poor	**-11.4±3.8**	-27.9±0.8	-17.3±4.1	-24.9±0.4	-25.9±0.9
	5 Trains	Good	-28.1±1.4	-6.4±0.6	-8.1±4.9	**-3.2±0.5**	-25.6±0.5
		Medium	-9.7±5.7	-17.9±1.0	-10.6±8.4	**-3.7±0.3**	-25.9±0.5
		Poor	-9.9±3.8	-24.7±1.9	**-9.5±6.6**	-11.1±4.0	-25.6±0.4
SMACv2	terran_5_vs_5	Replay	7.3±1.0	13.7±2.7	**13.8±4.4**	11.8±0.9	13.7±1.7
	zerg_5_vs_5	Replay	6.8±0.6	10.2±2.4	10.3±1.2	10.3±3.4	**10.6±0.7**
	terran_10_vs_10	Replay	7.4±0.5	10.4±2.5	12.7±2.0	11.8±2.0	**14.4±0.7**



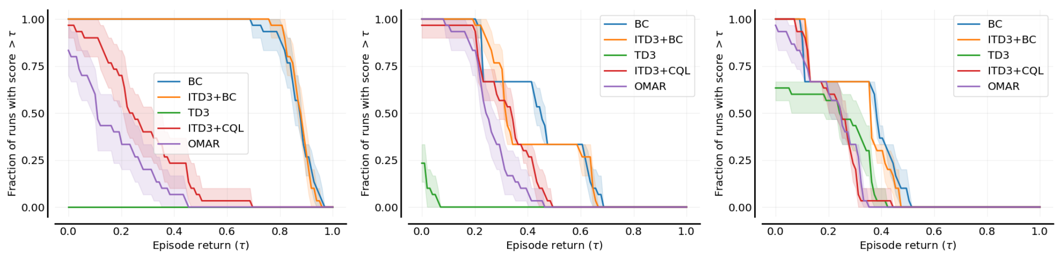
(a) SMAC Good (b) SMAC Medium (c) SMAC Poor

Figure D.1: Aggregated performance profiles (Agarwal et al., 2021) for SMAC. Shaded regions show pointwise 95% confidence bands based on percentile bootstrap with stratified sampling. Results were aggregated across all scenarios and seeds.

Continuous Actions. In Table D.5 we give the baseline results on datasets with continuous actions. In Figure D.2 we provide the aggregated performance profiles for MAMuJoCo.

Table D.5: Baseline results on datasets with continuous actions. The mean episode return with one standard deviation across all seeds is given. The best mean episode return in each row is given in bold.

Environment	Scenario	Dataset	BC	ITD3	ITD3+BC	ITD3+CQL	OMAR
MAMuJoCo	2-HalfCheetah	Good	6846±574	-578±33	**7025±439**	2934±1666	1434±1903
		Medium	1627±187	-87±223	**2561±82**	1755±283	1892±220
		Poor	465±59	-392±76	736±72	**739±191**	384±420
	2-Ant	Good	2697±267	-1274±501	**2922±194**	606±487	464±469
		Medium	**1145±126**	-1416±845	744±283	716±431	799±186
		Poor	954±80	741±398	**1256±122**	814±177	857±73
	4-Ant	Good	**2802±133**	-1033±432	2628±971	712±672	344±631
		Medium	1617±153	-1159±733	**1843±494**	1190±186	929±349
		Poor	1033±122	703±465	**1075±96**	518±122	518±112
PettingZoo	Pistonball	Good	**94.1±1.2**	0.8±10.6	93.1±2.0	-[2]	-
		Medium	10.3±6.0	-5.5±3.2	**13.7±8.9**	-	-
		Poor	4.6±3.2	-5.8±2.3	**6.1±2.5**	-	-
CityLearn	2022_all_phases	Replay	**-6576±39**	-6594±1	-6663±87	-6598±9	-6630±44
VoltageControl	case33_3min_final	Replay	**-9.9±2.3**	-10.0±0.8	-11.1±0.7	-32.9±6.7	-26.5±9.4



(a) MAMuJoCo Good (b) MAMuJoCo Medium (c) MAMuJoCo Poor

Figure D.2: Aggregated performance profiles (Agarwal et al., 2021) for MAMuJoCo. Shaded regions show pointwise 95% confidence bands based on percentile bootstrap with stratified sampling. Results were aggregated across all scenarios and seeds.

D.6 Reproducing Baseline Results

Scripts for reproducing our baseline experiments are included in the open-sourced code.

D.7 Baseline Compute Budget

To run all of our baselines we used CPUs on an internal compute cluster. In total we used 546 days of CPU compute time.

[2]Due to the nature of the CQL and OMAR algorithms, and the large number of agents in Pistonball we have not managed to successfully run these experiments without running out of RAM on the compute available to us. We are working on resolving this for the camera ready version.

E Dataset Licence, Author Statement, Hosting & Maintenance Plan

E.1 Dataset Licence

The datasets in OG-MARL are licenced under the Common Dataset Licences, **CC BY-NC-SA**.[3] This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format for noncommercial purposes only, and only so long as attribution is given to the creator. If you remix, adapt, or build upon the material, you must license the modified material under identical terms.

E.2 Author Statement

The authors of "Off-the-Grid MARL: Datasets with Baselines for Offline Multi-Agent Reinforcement Learning" bear all responsibility in case of any violation of rights during the collection of the data or other work, and will take appropriate action when needed, e.g. to remove data with such issues.

E.3 Hosting & Maintenance Plan

The OG-MARL datasets are hosted in an accessible, online storage bucket, kindly hosted by <anonymous organisation>. An easy-to-use interface for downloading datasets from the bucket is provided via our website. Datasets will continue to be maintained by the authors and dataset versions will be tracked on the OG-MARL GitHub repository. The OG-MARL code will also be tracked on GitHub. Issues and feature requests can be submitted on the GitHub repository.

Note: the OG-MARL GitHub repository is currently set to private during the anonymous review phase but will be made public as soon as the anonymous review is over. For now, all of the OG-MARL code is available for download from our anonymised website.

[3] https://paperswithcode.com/datasets/license